# Setup Guide | Config Shell Script

## Prerequisites

Ensure that the [SSD is already deployed](deploy_ssd.md) has already been checked|completed.

## **Overview**

This guide explains how to configure and set up the supplied (Power)Shell script for automated data submission within/from your Local Authority environment to a/the pre-agreed API endpoint. Descriptions and language used in this section unavoidably require some technical knowledge. It’s anticipated that after initial local testing, the API script will require inclusion into an LA’s overnights/cron job list. Towards this we have streamlined process(es) where possible to minimise both process(ing) overheads and run times.

## Shell Script Notes

### **Description**

The (Power)Shell script automates the process of extracting JSON structured Child Social Care records (payload) from an SSD defined table (data collection or staging table), sending the extracted payload to an API before updating appropriate submission status flags within the data collection or response table.

*Note: The use of MS Powershell within LA’s is commonplace. Thus we have assumed compatibility until such point that the wider pilot/early adopter group are able to contribute to the project understanding regarding localised restrictions or requirements.*

### **Key Features**

* Extracts pending JSON payload(s) from the SSD staging table(s)
* Sends JSON data to API endpoint or simulates the same process during localised initial testing.
* Updates submission status flags: Sent, Error, or Testing (ref: stored flags and settings are lowercase).

### **Parameters**

**DB/CMS Configuration details required for API**:

The API script, is stored and run from within your LA, and will need to be configured with some Local DB connection parameters :

* **$testingMode**: Bool flag to toggle btwn LA testing and data being sent externally. **true=NO data leaves the LA.**
* **$server**: CMS Reporting (SQL)Server name, e.g. ESLLREPORTS00X
* **$database**: Database instance where the SSD is deployed, e.g. for SystemC the default is : HDM\_Local

Some paramaters are pre-configured and requiring no local changes, here for reference:

* **$ssd-api\_data\_staging**: Add-on non-core SSD table containing JSON payloads and payload status information *(or ssd-api\_data\_staging\_anon during development and local testing)*
* **$url**: API endpoint path
* **$token**: Authentication token for the API

### **Prerequisites**

* PowerShell 5.1+ with ability to install SqlServer module.
* SQL Server with access to the specified database.

## **API Minimum Requirements**

### **System Requirements:**

* PowerShell: Version 5.1+ (check using $PSVersionTable.PSVersion) or alternative script language access.
* SQL Server: The CMS database is accessible and deployed SSD schema includes the additional ssd\_api\_data\_staging table.

### **Software Dependencies:**

SqlServer PowerShell Module: - This can be installed using: Install-Module -Name SqlServer -AllowClobber -Scope CurrentUser - Verify install: Get-Module -ListAvailable SqlServer

DB: - Ensure the ssd\_api\_data\_staging table has been created and populated as required.

## **Script Configuration**

Open the script in a text editor or Powershell and update the following variables to match your environment:

### **Connection details**

* $server = “YourSQLServerInstance” # E.g. “ESLLREPORTS00X”
* $database = “HDM\_Local” # E.g. default for SystemC/LLogic; update for Mosaic etc

### **API details**

* $url = “https://api.gov/endpoint” # API endpoint (already set)
* $token = “your-auth-token” # From client/API endpoint owner or supplied by D2I

### **Testing flag**

* $testingMode = $true # $true == **NO data leaves the LA.** | Set to $false for production-Live. See **Execution** for more detail.

## **Execution**

### Testing Mode:

* Ensure $testingMode = $true to simulate API calls without sending data.
* Run the script in PowerShell:
  + .\_json\_payload-sql-server-agent\_v0.2.4.ps1 (supplied filename may vary)

### Production Mode:

* After verifying functionality in testing, set $testingMode = $false.
* Re-run the script to send data to the API.

### Verify Updates:

* Check the ssd\_api\_data\_staging table to ensure submission\_status and api\_response are updated correctly.